Engenharia de Software - Parte 04

4 - ISO/IEC 9000-3

Há um conjunto de Normas da ISO desenvolvidas especificamente para *software*. O guia ISO/IEC 9000-3 aplica-se a empresas de *software* interessadas em estruturar seus Sistemas de Gestão da Qualidade com base nos requisitos da ISO 9001:2000.

4.1 SOFTWARE E QUALIDADE DE SOFTWARE

Poucas vezes se encontram na literatura definições claras para *software*. Convivemos com um suposto consenso sobre o seu significado: é como se todos soubessem o que é *software* e não fosse necessário defini-lo. Observam-se, no entanto, grandes dificuldades para a identificação de suas características próprias (tanto quanto do processo para seu desenvolvimento), o que pode ser atribuído, em boa medida, a esta indefinição.

Os conceitos de qualidade discutidos requerem um esforço de interpretação e adaptação para sua aplicação ao desenvolvimento e à manutenção de *software*, devido às características peculiares desse tipo de produto e de seu processo de desenvolvimento.

4.2 SOFTWARE E SUAS CARACTERÍSTICAS

Roger PRESSMAN define:
Software é:

- Instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados;
- Estruturas de dados que possibilitam que os programas manipulem adequadamente a informação; e
- Documentos que descrevem a operação e o uso dos programas." [PRE97]

Um dos aspectos de maior importância para o desenvolvimento de *software* de alta qualidade é a identificação das diferenças e das semelhanças entre produtos de *software* e produtos manufaturados. Há duas questões básicas:

1) *Software é diferente?*
2) *O processo de desenvolvimento de software é diferente?*

As respostas são *sim* e *não* para ambas as questões. Comecemos pelas diferenças. É preciso considerar que qualquer *software*, sendo um sistema lógico (e não físico), apresenta algumas características diferentes dos sistemas físicos (*hardware* de computador, por exemplo):

- *Software é desenvolvido, não manufaturado*. Seus custos estão concentrados em engenharia. Mesmo que levemos em consideração a crescente automação, através de ferramentas CASE e técnicas de quarta geração, devemos observar que ela se aplica sobretudo ao desenvolvimento. Alguns desenvolvimentos tecnológicos recentes trazem contribuições para redução desta diferenciação do *software* em relação à manufatura. As chamadas fábricas de *software* buscam estabelecer padrões de desenvolvimento e ferramentas adequadas para permitir a automação de determinadas etapas do processo. São contribuições nesta linha os geradores automáticos, os meta-sistemas, as bibliotecas de módulos prontos, os *plug-in´s*, os customizadores etc.

- *Software não se desgasta*. O número de falhas que apresenta em cada estágio de instalação está diretamente vinculado a erros do desenvolvimento, ou a alterações efetuadas (que podem acrescentar erros).

- *Software não dá "avisos" de que pode falhar mais tarde*. Em eletrônica, por exemplo, várias observações podem ser feitas periodicamente nos sinais para verificar o potencial de mau funcionamento.

- O reparo em software resulta, em geral, em nova definição do produto. No hardware, o reparo em geral é feito para restabelecer sua condição original.

- A maioria dos *softwares* é feita sob medida em vez de ser montada a partir de componentes existentes. A reutilização de componentes padrões de *software* ainda dá os seus primeiros passos.

A gerência do processo de *software* apresenta ainda algumas dificuldades características:

- Ainda faltam gerentes com experiência na área,

- Freqüentemente o *software* é o elemento do sistema que apresenta as funções ao usuário final, o que o torna mais exposto a mudanças,

- Muitos gerentes oriundos de outras áreas vêem *software* como uma arte e acabam por não aplicar sua experiência e instinto gerencial para resolver os problemas de *software*.

Passemos agora às semelhanças. Todas diferenças expostas reforçam uma semelhança básica com a manufatura — a necessidade de forte gerência do processo — através de:

- Planos detalhados,

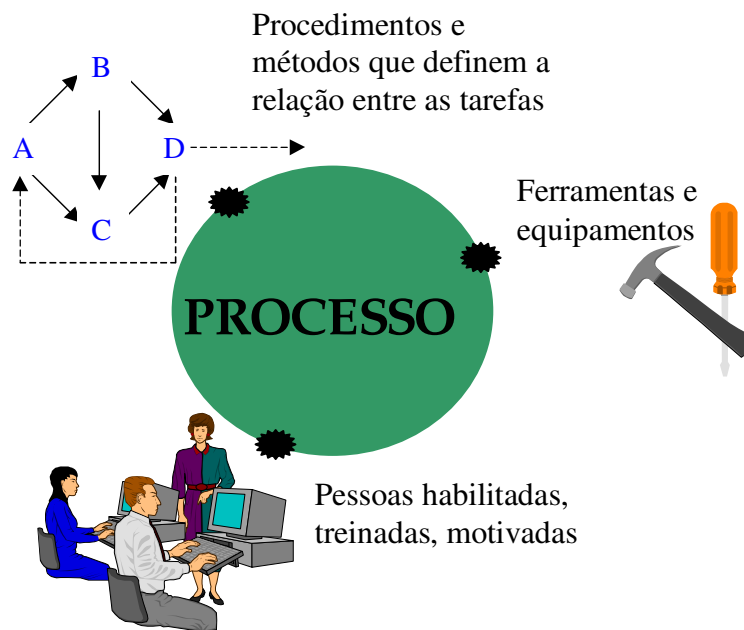
- Estimativas,

- Medições,

- Avaliações,

- Controle.

As atividades técnicas — análise e projeto, algoritmos, estruturação de dados, linguagens etc. — devem ser tratadas com métodos, técnicas e ferramentas específicos, mas a gerência do processo como um todo precisa se utilizar de conceitos e métodos idênticos aos de outras áreas: a definição de objetivos claros para a organização, o comprometimento da alta administração, o envolvimento e o preparo das pessoas, a sincronização de ações em uma mesma direção. As ferramentas para avaliação e controle gerencial também são as mesmas.

Compreendidas as características do *software*, passaremos a conceituar produto e processo de *software*, criando condições para uma conceituação ampla de qualidade de *software*, que envolva esses dois aspectos.

Processo e seus componentes

4.3 QUALIDADE DE SOFTWARE

O que significa qualidade de *software*? Uma definição é proposta por R. PRESSMAN:

> *"Conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente definido." [PRE97]*

Esta definição tem o mérito de considerar, além dos requisitos explícitos, também a adequação aos requisitos implícitos, comumente presentes. No entanto, está longe de representar um conceito geral de qualidade de *software*.

Independentemente da resposta que se dê à pergunta acima formulada, é sempre relativo o valor desta resposta, como mostra a tabela seguir baseada em estudo de Weinberg [WEI93].

As pessoas por trás das definições de qualidade

QUEM É A PESSOA POR TRÁS DESSA DEFINIÇÃO DE QUALIDADE?	
Defeito zero	• Para os usuários cujo trabalho é afetado pelos defeitos. • Para os gerentes que são criticados pelos defeitos.
Ter um grande número de funções	• Para os usuários cujo trabalho pode tirar proveito dessas funções — se eles as conhecerem. • Para os distribuidores que acreditam que as funções vendem produtos.
Codificação elegante	• Para o pessoal de desenvolvimento que dá um grande valor às opiniões de seus colegas. • Para os professores de ciência da computação que apreciam elegância.
Alto desempenho	• Para os usuários cujo trabalho sobrecarrega a capacidade de suas máquinas. • Para o pessoal de venda que tem de submeter seus produtos a benchmarks.
Baixo custo de desenvolvimento	• Para os usuários que desejam comprar milhares de cópias do *software*. • Para os gerentes de projeto que estão com orçamentos apertados.
Desenvolvimento rápido	• Para os usuários cujo trabalho está esperando pelo *software*. • Para os distribuidores que desejam colonizar um mercado antes de seus concorrentes.
Facilidade para o usuário	• Para os usuários que gastam oito horas por dia na frente de uma tela utilizando o *software*. • Para os usuários que não conseguem se lembrar de detalhes de interface.

WEINBERG mostra que diferentes pessoas podem ter percepções diferentes em relação à qualidade de um mesmo produto de *software*. Ele levantou algumas definições potenciais para *alta qualidade de software* e identificou os tipos de pessoas que podem estar por trás de cada uma delas.

A *Gestão da Qualidade Total (TQM)* em empresas de desenvolvimento de *software* possui o mesmo significado que nas empresas de manufatura e de serviços.

Tendo como pontos de apoio — estratégicos — a busca da satisfação do cliente, o posicionamento consciente em relação ao mercado e a melhoria contínua, desdobram-se a política e os objetivos da organização. Esses objetivos são continuamente reavaliados e aperfeiçoados. A partir desses objetivos, articulam-se as diversas ações necessárias para atingi-los e sincronizam-se as atividades nos diversos níveis, desde a Alta Direção até cada um dos postos de trabalho.

Na gerência dos seus processos a empresa que produz *software* necessita mover-se na direção da *prevenção de defeitos*, em oposição à antiga postura de *detecção e remoção de defeitos*, que via a qualidade como função da inspeção final (testes) dos programas.

A melhoria de cada um dos processos pode ser obtida aplicando-se o ciclo *Planeje / Faça / Verifique / Aja (PDCA — Plan / Do / Check / Act)* também conhecido como ciclo de Deming. No nível operacional o PDCA constitui um método eficiente e direcionado para resolução dinâmica dos problemas.

O amadurecimento da relação com o cliente e a busca de sua satisfação são aspectos focalizados na Qualidade Total. No desenvolvimento de *software* o contrato formal entre cliente e fornecedor adquire grande importância.

A busca de alta qualidade em *software* tem sido realizada através de dois tratamentos principais:

- *Gerência do processo*, que considera a necessidade de estabelecer um processo de qualidade para produzir produtos de qualidade,

- *Gerência do produto*, que busca identificar as características tangíveis dos produtos a serem desenvolvidos, estabelecendo a partir delas diretrizes para o processo de desenvolvimento.

Exemplos recentes têm demonstrado que ambas visões têm suas contribuições e levam a resultados de melhoria. A *AT&T*, por exemplo, focalizou a melhoria de seus produtos, enquanto que a *Motorola* utilizou a melhoria de processos. Em ambas visões é crescente o tratamento objetivo da gerência, com o uso de métricas de processo e de produto.

A discussão processo x produto de *software*, fortemente presente hoje na literatura de engenharia de *software*, contribui para identificarmos as vantagens e desvantagens de cada um dos tratamentos e concluirmos que um deles por si só não preenche todas as necessidades de melhoria no desenvolvimento de *software*.

4.4 QUALIDADE DE PROCESSO DE SOFTWARE

A exemplo do que ocorre há anos na manufatura, a gerência do processo de *software* e a busca da sua melhoria continua apresentam-se como práticas necessárias para o melhor desempenho econômico das empresas.

A gerência do processo visa estabelecer uma infra-estrutura para suportar e guiar os trabalhos dos diversos projetos de maneira uniforme. A gerência inclui:

- *Definição do processo*: estabelecer um padrão para implementação, avaliação e melhoria de cada tarefa;

- *Execução do processo*: definir os métodos e técnicas usadas para produzir produtos com qualidade;

- *Coleta de dados e análise*: tratar as medições realizadas dos produtos e processos de *software*, e o uso desses dados;

- *Controle do processo*: estabelecer mecanismos para certificar o desempenho do processo definido; monitorar e ajustar o processo onde melhorias forem necessárias.

Estudos e trabalhos realizados pelo *SEL — Software Engineering Laboratory*, um esforço cooperativo da *NASA*, através da *Divisão de Dinâmica de Vôo* (*Flight Dynamics Division — FDD*), com o *Departamento de Ciência da Computação da Universidade de Maryland* — demonstraram a importância da gerência do processo para obtenção de padrões cada vez mais elevados de qualidade.

Vitor BASILI et al. informam que o SEL investiu, durante duas décadas, aproximadamente 11% de seu orçamento na melhoria do processo. Os principais resultados obtidos na definição dos processos do SEL foram [BAS95]:

- A *institucionalização* da mudança e da melhoria do processo como práticas de negócio padrões,
- O uso intensivo de *medições*: elas são esperadas, aplicadas e efetivas, tendo se tornado uma maneira de fazer negócio,
- O direcionamento das mudanças para *processo e produto*
- *Mudança bottom-up*, baseada, em grande medida, na experiência do pessoal em desenvolvimento de projetos,
- Maior ênfase a *tecnologias orientadas a pessoas* (*people-oriented* — revisões, inspeções, técnicas de sala limpa etc.) que a automação.

O processo *sala limpa* (*cleanroom*) recebeu no *SEL* um tratamento baseado na experiência adquirida em seus projetos e tem como uma de suas bases a ênfase em técnicas de leitura. Foi definido com quatro elementos:

- Separação entre as equipes de desenvolvimento e testes
- Dependência maior de revisões dos pares (*peer reviews*) do que de testes de unidades como primeira técnica de verificação do desenvolvedor (BASILI e GREEN, 1994, p. 58)
- Uso de máquinas de estado e funções informais para definir o projeto do sistema
- Um tratamento estatístico para testes baseados em cenários operacionais.

Chama a atenção, entre as conclusões do *SEL*, a crescente consideração do fator humano na melhoria dos processos, contrariando a expectativa de tornar os processos independentes das pessoas, uma visão necessária para a definição clara de procedimentos e ferramentas, mas que se mostra limitada diante da constatação de que um processo definido necessita ainda mais fortemente das pessoas preparadas para implementá-lo.

4.5 ISO/IEC 9000-3 – APLICAÇÃO DA ISO 9001:2000 A SOFTWARE

A ISO/IEC 9000-3 (*Software and Systems Engineering — Guidelines for the Application of ISO 9001:2000 to Computer Software*) é uma Norma da ISO que fornece um guia para organizações na aplicação da ISO 9001:2000 para aquisição, fornecimento, desenvolvimento, operação e manutenção de *software* de computador.

Não é usada como critério de avaliação em certificação/registro de Sistema de Gestão da Qualidade, pois não adiciona ou muda os requisitos da ISO 9001.

O guia ISO/IEC 9000-3 foi preparado pelo Comitê Técnico ISO/IEC/TC JTC1, Subcomitê SC 7/WG18, *Software and Systems Engineering*, com representantes do ISO TC176 SC 2. A ISO colabora com a Comissão Eletrotécnica Internacional (*International Electrotechnical Commission - IEC*) em todos os assuntos referentes a padrões eletrotécnicos.

A mostra as aplicações para as quais a ISO/IEC 9000-3 é apropriada. [ISO02, 1.2]

Aplicações da ISO/IEC 9000-3

TIPO DE *SOFTWARE*
Parte de contrato comercial com outra organização
Produto disponível para um setor de mercado
Usado para apoiar processos de uma organização
Embutido (embarcado) em um produto de *hardware*
Relacionado a serviços de *software*

A aplicação da ISO/IEC 9000-3 independe de tecnologia, modelos de ciclo de vida, processos de desenvolvimento, seqüência de atividades ou estrutura organizacional da organização. Nas organizações cuja abrangência de atividades que inclui áreas diferentes de desenvolvimento de *software*, o relacionamento entre elementos de *software* do seu sistema de gestão da qualidade e os demais aspectos devem claramente ser documentados no sistema de gestão da qualidade como um todo.

As organizações com Sistemas de Gestão da Qualidade para desenvolvimento, operação e manutenção de *software* baseados na ISO/IEC 9000-3 podem escolher usar processos da ISO/IEC 12207 para suportar ou complementar o modelo de processo da ISO 9001:2000. Os parágrafos relacionados à ISO/IEC 12207 são referenciados em cada seção da ISO/IEC 9000-3, mas não se pretende com isso incluir requisitos adicionais aos da ISO 9001:2000.